

Zabbixを もっと使おう

2016/11/06 @OSC 2016 TOKYO/FALL

ZABBIX-JP 田中 敦

自己紹介

- ▶ 氏名
 - ▶ 田中 敦 (たなか あつし)
- ▶ 所属
 - ▶ ZABBIX-JP (Zabbixの日本のコミュニティ)
- ▶ 役職
 - ▶ 副代表
- ▶ 主な活動
 - ▶ コミュニティサイト (<http://www.zabbix.jp/>) にて、情報発信と問い合わせ対応
 - ▶ ZabbixのUIなどの日本語翻訳

アジェンダ

- ▶ はじめに
 - ▶ Zabbixの振り返り
- ▶ 監視方法の詳細
 - ▶ 標準の機能
 - ▶ 機能の拡張
- ▶ 応用例
- ▶ Zabbix 3.2について
- ▶ まとめ

はじめに

Zabbixとは

- ▶ ラトビアにあるZabbix LLCという会社が開発しているオープンソースの統合監視ツール
- ▶ ライセンスは、GPLv2
- ▶ Webブラウザを使用して監視の設定ができる
- ▶ 専用のエージェント、SNMP、IPMI、JMX、SSH、Telnet、独自のスクリプトなど様々な方法を利用して監視が可能
- ▶ 取得した値をグラフ化したり、マップと呼ばれる機能で障害発生個所を把握しやすい
- ▶ テンプレートと呼ばれる監視設定のセットを事前に用意しておくことで、新規に追加されたサーバーの監視を速やかに開始できる

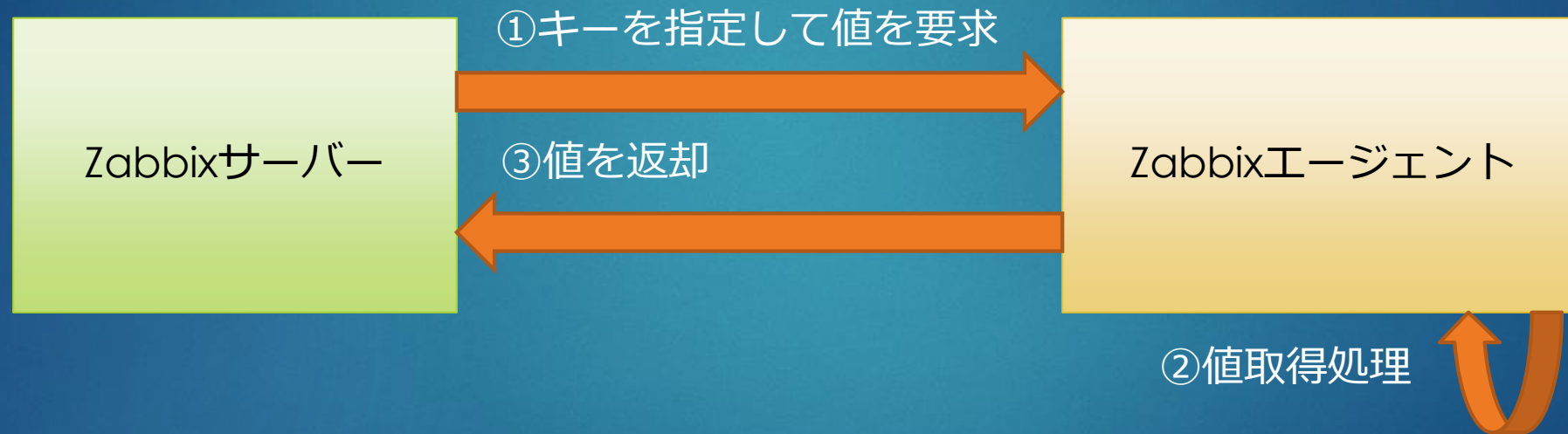
Zabbix用語の振り返り

用語	概要
アイテム	監視する項目のこと CPU、メモリ、ネットワーク、ハードディスク、ログファイルなど、数値や文字列で値を取得する
トリガー	アイテムで取得してきた値に対して、条件を付与して障害かどうかを判定する条件式を設定するもの
アクション	トリガーなどでイベントが発生したときにメールでの通知やスクリプトの実行などを行う設定のこと
テンプレート	アイテム、トリガー、グラフなどをひとまとまりにして、監視対象を追加したときに、これを紐づけるだけで監視を開始できるようにする

監視方法の詳細

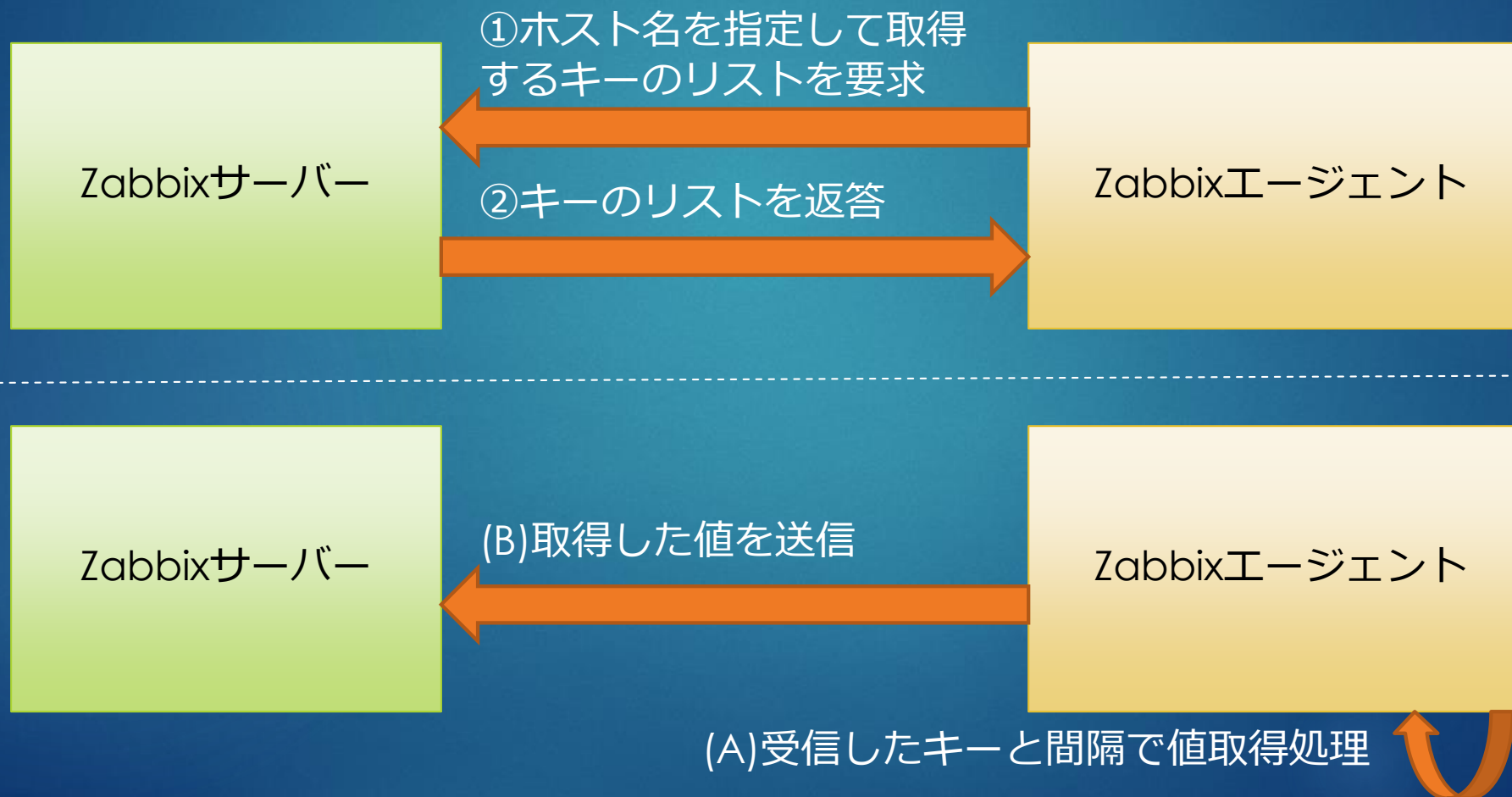
監視項目の取得シーケンス（1）

アイテムのタイプが、Zabbixエージェントの場合



監視項目の取得シーケンス（2）

アイテムのタイプが、Zabbixエージェント（アクティブ）の場合



簡単な拡張 (UserParameter)

10

- ▶ 標準で用意されているアイテムのキー以外にも、設定ファイルにUserParameterという設定を追加することで、標準以外の項目も監視することができる
- ▶ Zabbixサーバーとのやり取りは、通常のZabbixエージェントと同様
- ▶ 例：
`/etc/zabbix/zabbix_agentd.d/userparameter_mysql.conf`

UserParameterの例

11

- ▶ パッケージでのインストール時にデフォルトで用意されている設定例

```
UserParameter=mysql.status[*],echo "show global status where Variable_name='$1';" |  
HOME=/var/lib/zabbix mysql -N | awk '{print $$2}'
```

- ▶ キー（引数あり）

```
mysql.status[*]
```

- ▶ コマンド文字列

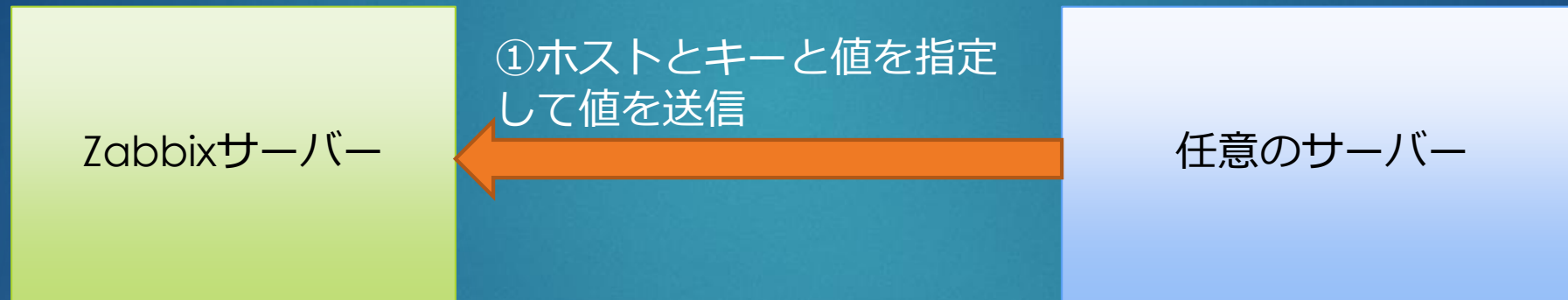
```
echo "show global status where Variable_name='$1';" |  
HOME=/var/lib/zabbix mysql -N | awk '{print $$2}'
```

簡単な拡張（トラッパー）

- ▶ 別の拡張方法として、トラッパーと呼ばれるものを使用できる
- ▶ アイテムのタイプとして「Zabbixトラッパー」を選択して作成しておくことで、zabbix_senderというコマンドを使用して、アイテムの値を任意のタイミングでZabbixサーバーに通知することができる
- ▶ Zabbixエージェントをインストールできないサーバーから定期的に値の通知だけを行わせることにも使用できる
- ▶ システムリソースのメトリクスだけではなく、各種センサーの値や業務システムのメトリクスをZabbixサーバーに集約することも可

Zabbixトラッパーのシーケンス

13



Zabbixトラッパーの応用例

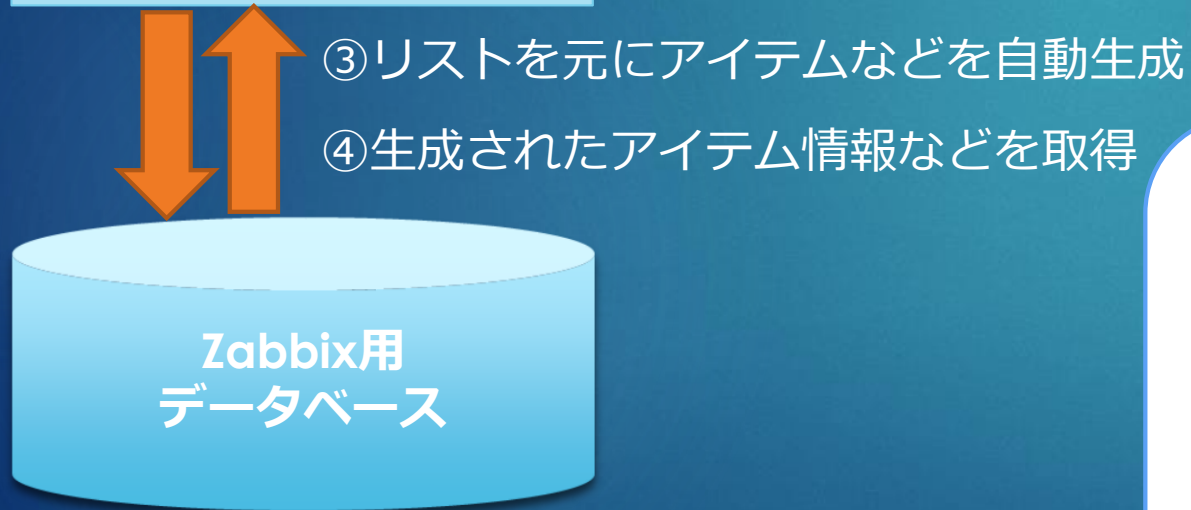
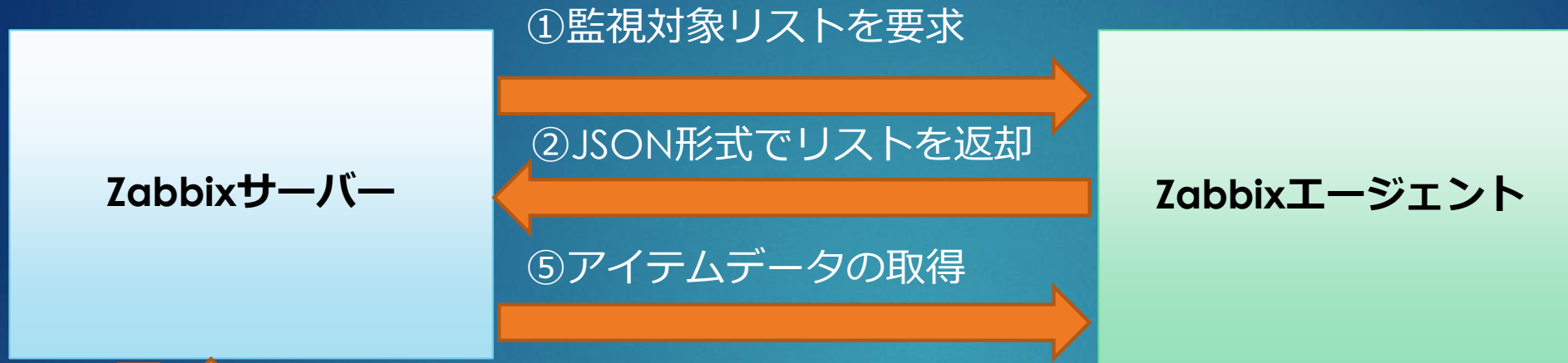
14

- ▶ ジョブの終了ステータスの通知
 - ▶ ジョブの終了時に、正常終了か異常終了かをZabbixサーバーに通知
 - ▶ 通知された値によって、Zabbixのトリガーで判定して運用管理者にメールなどで通知
- ▶ 業務システムなどの指標値の視覚化
 - ▶ アンケートシステムでのアンケート収集状況を視覚化
 - ▶ 登録ユーザーの属性に合わせて分類して視覚化
 - ▶ IoTの各種デバイスやセンサーの値を収集して監視

- ▶ ローレベルディスクカバリ (LLD)
 - ▶ ネットワークインターフェースの自動認識
 - ▶ マウントされたハードディスクの自動認識
 - ▶ SNMP対応機器のインターフェースの自動認識

標準で用意されているLLDの例

16



```
{  
  "data": [  
    {"#FSNAME": "/", "#FSTYPE": "rootfs"},  
    {"#FSNAME": "/proc", "#FSTYPE": "proc"},  
    {"#FSNAME": "/sys", "#FSTYPE": "sysfs"},  
    {"#FSNAME": "/", "#FSTYPE": "ext4"},  
    {"#FSNAME": "/proc/bus/usb", "#FSTYPE": "usbfs"},  
    {"#FSNAME": "/boot", "#FSTYPE": "ext4"},  
  ]  
}
```


さらなる拡張

- ▶ 独自LLDの利用
 - ▶ 標準機能で用意されたLLDの機能以外にも、独自のディスクバリ用のスクリプトなどを用意することで、独自の動的に数が変化する監視対象を自動認識
- ▶ 独自LLDの応用例
 - ▶ PostgreSQL monitoring template for Zabbix (pg_monz)
- ▶ APIの利用

拡張時の注意点

- ▶ 1つの値取得処理にかかる時間に注意すること
 - ▶ デフォルトの設定では、Zabbixサーバー側もZabbixエージェント側も3秒間でタイムアウトが発生してしまうので、ZabbixサーバーやZabbixエージェントが値を取得する処理に時間がかかるようであれば、設定ファイルのTimeoutの値を長めに設定する（最大値は30秒）
 - ▶ リモートからクラウド環境を監視するような場合でも、値取得に時間がかかるなら同様
 - ▶ 30秒以上かかるような処理は、別途、結果をファイルに書き出して、Zabbixがそのファイルから読み出すようにするか、トラッパーを利用して、zabbix_senderコマンドでZabbixサーバーに通知するようにする

応用例

動的に監視対象の数が変化する環境の監視

- ▶ クラウド環境
 - ▶ AWS環境の監視
 - ▶ CloudWatchのメトリクスをZabbix上で障害判定と長期保管
- ▶ データベース
 - ▶ PostgreSQLの監視
 - ▶ DBMS上に登録されている複数のデータベースの情報を自動的に取得して監視項目の自動設定 (ex. pg_monz、LLDやトラッパーを利用)

他のシステムやプロダクトとの連携

21

- ▶ ログの分析
 - ▶ Fluentdの出力にZabbixを指定することで、Fluentdのソースから取得した情報を集計など加工して監視できるようにする (ex. fluent-plugin-zabbix、トラッパー利用)
- ▶ 視覚化の拡張
 - ▶ デフォルトのグラフ機能よりも様々な表現力のあるグラフを生成する (ex. Zabbix plugin for Grafana、API利用)
- ▶ インシデント管理
 - ▶ RedmineやOTRSとの連携 (リモートコマンドやメディアスクリプト利用)

Zabbix plugin for Grafana

22



<https://github.com/alexanderzobnin/grafana-zabbix>

Zabbix 3.2情報

Zabbix 3.2リリース

- ▶ Zabbix 3.2が2016年9月14日にリリースされた。
 - ▶ 現時点の最新版は、Zabbix 3.2.1
- ▶ Zabbix 3.2は、ポイントリリースと呼ばれるバージョン
 - ▶ サポート期間が短いことに注意
 - ▶ 次のバージョンがリリースされたら、約1ヶ月でサポートが終了する予定
 - ▶ 長期間のサポートが必要なのであれば、長期間のサポートが用意されているLTSと呼ばれるバージョン（例：Zabbix 3.0）を選択すること

Zabbix 3.2の新機能と変更点

25

- ▶ トリガーの手動クローズ
- ▶ イベントタグの設定
- ▶ イベント相関関係の設定
- ▶ ヒステリシストリガーの改善
- ▶ LLDで生成したアイテムの詳細確認
- ▶ Webシナリオのエクスポートとインポート
- ▶ イベント画面から障害画面へ
- ▶ ホストグループの階層化

トリガーの手動クローズ

- ▶ これまで、ログ監視やSNMPトラッパーの監視で、トリガーの状態が「障害」のままとなつて、無理矢理、関数nodata()を使用して一定時間後にトリガーの状態をクリアしていた
- ▶ これでは、対処しなくてもトリガーの状態が「正常」に戻ってしまうので、本当にその検知した障害に対応できたのかがわからない
- ▶ Zabbix 3.2では、トリガーの設定として、手動でクローズすることができるかどうかを設定できるようになった

トリガーの設定画面

The image shows the Zabbix web interface for configuring a trigger. The main page is titled 'トリガー' (Trigger) and shows configuration options for a trigger named 'Zabbix server'. The configuration includes fields for name, depth, and conditions, as well as buttons for '正常イベントの生成' (Generate normal event), '障害イベント生成モード' (Incident event generation mode), and '正常時のイベントクローズ' (Close event in normal state). A checkbox labeled '手動でのクローズ許可' (Allow manual close) is circled in red.

A modal dialog titled '障害対応コメント入力' (Incident Response Comment Input) is open, showing a text area for 'メッセージ' (Message) and a '履歴' (History) table. The dialog also has radio buttons for '障害対応コメント' (Incident response comment) and a checkbox for '障害のクローズ' (Close incident), both of which are circled in red. The dialog includes buttons for '障害対応コメント' (Incident response comment) and 'キャンセル' (Cancel).

ヒステリシストリガーの改善

- ▶ いままでのトリガーの条件設定は、条件式に合致するかしないかしか指定ができなかった
- ▶ 障害が発生している状態で、回復したとみなす閾値を別途設定するために、ヒステリシストリガーと呼ばれる設定があった
 - ▶ 例

```
{TRIGGER.VALUE}=0 and {server:system.cpu.load[,avg1].last()}>10) or  
{TRIGGER.VALUE}=1 and {server:system.cpu.load[,avg1].last()}>5)
```

- ▶ Zabbix 3.2では、回復したと判定する方法として、別の条件式を設定することも可能となった

トリガーの条件式と復旧条件式

29

ZABBIX 監視データ インベントリ レポート 設定 管理

ホストグループ テンプレート ホスト メンテナンス アクション イベント 相関関係 ディスカバリ ITサービス

トリガー

すべてのホスト / Zabbix server 有効 ZBX SNMP JMX IPMI アプリケーション 11 アイテム 78 トリガー 47

トリガー 依存関係

名前

深刻度 未分類 情報 警告 軽度の障害 重度の障害 致命的な障害

条件式

[条件式ビルダー](#)

正常イベントの生成 条件式 復旧条件式 なし

障害イベントの生成 なし 複数

正常時のイベントクローズ すべての障害 タグの値が一致したすべての障害

タグ タグ 値 [削除](#)

[追加](#)

ZABBIX 監視データ インベントリ レポート 設定 管理

ホストグループ テンプレート ホスト メンテナンス アクション イベント 相関関係 ディスカバリ ITサービス

トリガー

すべてのホスト / Zabbix server 有効 ZBX SNMP JMX IPMI アプリケーション 11 アイテム 78 トリガー 47 グラフ 14 ディス

トリガー 依存関係

名前

深刻度 未分類 情報 警告 軽度の障害 重度の障害 致命的な障害

障害の条件式

[条件式ビルダー](#)

正常イベントの生成 条件式 復旧条件式 なし

復旧条件式

[条件式ビルダー](#)

[追加](#)

Zabbix 3.2アップグレード時留意点 1

30

- ▶ RHEL 7やCentOS 7でsystemdでzabbix-serverを起動している場合
 - ▶ 3.2.xにソフトウェアをアップグレードし、最初にzabbix-serverを起動する時は、systemctlコマンドで起動しないこと
 - ▶ zabbix_serverプロセス起動時のデータベースアップグレード処理に時間がかかってしまう場合がある
 - ▶ 次ページのようにコマンドで手動起動し、ログファイルを確認してデータベースのアップグレードが正常に終了したことを確認しておくこと

※詳細 : ZBX-11203 database upgrade from Zabbix 3.0 to Zabbix 3.2 fails
<https://support.zabbix.com/browse/ZBX-11203>

Zabbix 3.2アップデート後の初期起動

▶ コマンド実行例

```
# sudo -u zabbix /usr/sbin/zabbix_server -f -c /etc/zabbix/zabbix_server.conf
```

▶ ログサンプル

```
30488:20160916:103655.733 current database version (mandatory/optional): 02040000/02040000
30488:20160916:103655.733 required mandatory version: 03020000
30488:20160916:103655.733 starting automatic database upgrade
30488:20160916:103655.836 completed 1% of database upgrade
30488:20160916:103655.852 completed 2% of database upgrade

30488:20160916:103657.732 completed 97% of database upgrade
30488:20160916:103657.736 completed 98% of database upgrade
30488:20160916:103657.737 completed 100% of database upgrade
30488:20160916:103657.737 database upgrade fully completed
```

Zabbix 3.2アップグレード時留意点 2

32

- ▶ ODBCを使用したデータベースアクセス（データベースモニタ）のアイテムが正常に取得できない。
- ▶ 以下のようなログが記録される。

```
20850:20160915:160911.863 error reason for “ホスト名:db.odbc.select[クエリ名]” changed:  
Received value [??p] is not suitable for value type [Numeric (unsigned)] and data type  
[Decimal]
```

- ▶ 次バージョン（3.2.2）で修正予定。

※詳細 : ZBX-11213 DB Monitoring stopped working after upgrade to 3.2
<https://support.zabbix.com/browse/ZBX-11213>

RHEL 6/CentOS 6への導入

- ▶ Zabbix 3.0以降では、RHEL 6/CentOS 6は、Zabbixサーバーを構築する環境としては非推奨となった
 - ▶ Webインターフェースの稼働条件として、PHP 5.4以降のみサポートであるため、RHEL 6/CentOS 6の標準のPHPのバージョンが古いので対応不可
 - ▶他にも一部ライブラリのバージョンが古いため、Zabbixサーバーの新機能が利用できないものがある（例：SMTP認証）
- ▶ 対策として、RHEL 7/CentOS 7に移行するか、一部機能の制限があるものの、SCL、サードベンダーのパッケージ、ソースからの導入などで、新しいバージョンのPHPやライブラリを使用してZabbixサーバーやZabbixのWebインターフェースを稼働させることはできる

RHEL 6へのインストールヒント

34

```
# yum install centos-release-scl
# yum install rh-php56-php rh-php56-php-bcmath rh-php56-php-gd rh-php56-php-ldap rh-php56-php-mbstring rh-php56-php-xml rh-php56-php-mysqlnd
# yum install http://repo.zabbix.com/zabbix/3.2/rhel/6/x86_64/zabbix-release-3.2-1.el6.noarch.rpm
# yum install zabbix-agent
# yum --enablerepo=zabbix-deprecated install zabbix-server-mysql
# yum --enablerepo=zabbix-deprecated install zabbix-web-mysql zabbix-web-japanese
# yum install mysql-server
# vi /etc/my.cnf
# service mysqld start
# chkconfig mysqld on
# mysql -uroot
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'password' ;
mysql> exit
# zcat /usr/share/doc/zabbix-server-mysql-3.2.1/create.sql.gz | mysql -uroot zabbix
# vi /etc/zabbix/zabbix_server.conf
# cp /usr/share/doc/zabbix-web-3.2.1/httpd24-example.conf /opt/rh/httpd24/root/etc/httpd/conf.d/zabbix.conf
# vi /opt/rh/httpd24/root/etc/httpd/conf.d/zabbix.conf
# setsebool -P httpd_can_network_connect on
# setsebool -P zabbix_can_network on
# service httpd24-httpd start
# service zabbix-agent start
# service zabbix-server start
# yum install policycoreutils-python
# grep zabbix_server /var/log/audit/audit.log | audit2allow -M zabbix-limit
# semodule -i zabbix-limit.pp
# service zabbix-server start
```

まとめ

Zabbixの機能を活用することで

- ▶ 拡張や他システムとの連携によって
 - ▶ より広範囲の情報を把握して視覚化できる
 - ▶ 運用管理の担当者の監視設定の負荷を軽減する
 - ▶ 運用の効率化と継続的改善へ

Zabbix 3.2以降の新機能で

- ▶ トリッキーな設定ではなく、標準の機能として確認済みのトリガーを手動で確認済みの状態に戻すことができる
- ▶ より柔軟なトリガーの依存関係を設定して、自動的に状態をリセットし、本来関知して対応しなければならない障害に注力しやすくなる

- ▶ PostgreSQL monitoring template for Zabbix (pg_monz)
http://pg-monz.github.io/pg_monz/
- ▶ fluent-plugin-zabbix
<http://sfujiwara.hatenablog.com/entry/20120417/1334640635>
<https://github.com/fujiwara/fluent-plugin-zabbix>
- ▶ Zabbix plugin for Grafana
<https://github.com/alexanderzobnin/grafana-zabbix>
- ▶ Zabbix 3.2新機能
http://www.zabbix.com/jp/whats_new

▶ ご清聴いただきありがとうございました。

- ✓ 本資料内で利用させて頂いた各プロダクト名やサービス名などは、各社もしくは各団体の商標または登録商標です。